SPYWOLF

Security Audit Report

Completed on

August 03, 2022

MADE IN USA 🇺🇸

OVERVIEW

This audit has been prepared for **JAILKWON** to review the main aspects of the project to help investors make make an informative decision during their research process.

You will find a a summarized review of the following key points:

- ✔ Contract's source code
- ✔ Owners' wallets
- ✔ Tokenomics
- ✔ Team transparency and goals
- ✔ Website's age, code, security and UX
- ✔ Whitepaper and roadmap
- ✔ Social media & online presence

The results of this audit are purely based on the team's evaluation and does not guarantee nor reflect the projects outcome and goal

- SPYWOLF Team -

TABLE OF CONTENTS

JAILKWON

PROJECT DESCRIPTION

According to their website:

JKWON was founded by a group of disgruntled LUNA investors. Almost immediately after the Luna crash, the Jailkwon project became very popular and a number of community members approached Jailkwon Token with a desire to help.

Future developments of the project:

- JKWON wallet
- JKWON swap
- NFT marketplace

Release Date: Presale starts on August, 2022

Category: Meme

CONTRACT INFO

Token Name	Symbol
JAILKWON	$JKWON

Contract Address
0x24e537c31D18c926d96616898C62DF93a4a5c7E6

Network	Language
Binance Smart Chain	Solidity

Deployment Date	Verified?
August 02, 2022	Yes

Total Supply	Status
50,000,000,000	Not launched

TAXES

Buy Tax	Sell Tax
5%	5%

*Taxes can be changed in future

Our Contract Review Process

The contract review process pays special attention to the following:

- Testing the smart contracts against both common and uncommon vulnerabilities
- Assessing the codebase to ensure compliance with current best practices and industry standards.
- Ensuring contract logic meets the specifications and intentions of the client.
- Cross referencing contract structure and implementation against similar smart contracts produced by industry leaders.
- Thorough line-by-line manual review of the entire codebase by industry experts.

Blockchain security tools used:

- OpenZeppelin
- Mythril
- Solidity Compiler
- Hardhat

CURRENT STATS

(As of August 03, 2022)

Liquidity

Not added yet

Burn

No burnt tokens

Status:

Not Launched!

MaxTxAmount

500,000,000

DEX:

PancakeSwap

LP Address(es)

Liquidity not added yet

TOKEN TRANSFERS STATS

Transfer Count	2
Uniq Senders	2
Uniq Receivers	2
Total Amount	99999999999.99998 $JKWON
Median Transfer Amount	49999999999.99999 $JKWON
Average Transfer Amount	49999999999.99999 $JKWON
First transfer date	2022-08-02
Last transfer date	2022-08-02
Days token transferred	1

SMART CONTRACT STATS

Calls Count	3
External calls	3
Internal calls	0
Transactions count	3
Uniq Callers	1
Days contract called	1
Last transaction time	2022-08-02 18:19:02 UTC
Created	2022-08-02 18:10:08 UTC
Create TX	0x69985cd32203cfcca7a5329c18146413f2d38c8a1dccf3cb351028bcd4f24999
Creator	0x7e5f30521b12ca7025e5b24db7fa6e51bf3ee81a

FEATURED WALLETS

Owner address	0xd9b8fa41e2b7e4414631ff722ec9c7ef24251453
Reserve wallet	0xb280d1372d06d42c6d40f328e3afafabfec950e9
Reward distributor	Same as owner
LP address	**Liquidity not added yet**

TOP 3 UNLOCKED WALLETS

1

100% Same as owner

*Tokens are not distributed yet

VULNERABILITY CHECK

Design Logic	Passed
Compiler warnings.	Passed
Private user data leaks	Passed
Timestamp dependence	Passed
Integer overflow and underflow	Passed
Race conditions and reentrancy. Cross-function race conditions	Passed
Possible delays in data delivery	Passed
Oracle calls	Passed
Front running	Passed
DoS with Revert	Passed
DoS with block gas limit	Passed
Methods execution permissions	Passed
Economy model	Passed
Impact of the exchange rate on the logic	Passed
Malicious Event log	Passed
Scoping and declarations	Passed
Uninitialized storage pointers	Passed
Arithmetic accuracy	Passed
Cross-function race conditions	Passed
Safe Zeppelin module	Passed
Fallback function security	Passed

THREAT LEVELS

When performing smart contract audits, our specialists look for known vulnerabilities as well as logical and access control issues within the code. The exploitation of these issues by malicious actors may cause serious financial damage to projects that failed to get an audit in time. We categorize these vulnerabilities by the following levels:

High Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Medium Risk

Issues on this level are critical to the smart contract's performance/functionality and should be fixed before moving to a live environment.

Low Risk

Issues on this level are minor details and warning that can remain unfixed.

Informational

Information level is to offer suggestions for improvement of efficacy or security for features with a risk free factor.

FOUND THREATS

⚠ High Risk

Owner can set buy/sell fees up to 75%.

```
uint256 private constant _MAX_FEE_LIMIT = 25;

function changeFeesForNormalBuy(uint8 _liquidityFeeOnBuy, uint8 _marketingFeeOnBuy, uint8 _EtrondistributionFeeOnBuy) external onlyOwner {
    require(_liquidityFeeOnBuy <= _MAX_FEE_LIMIT, "LiquidityFeeOnBuy can't exceed 25%!");
    require(_marketingFeeOnBuy <= _MAX_FEE_LIMIT, "MarketingFeeOnBuy can't exceed 25%!");
    require(_EtrondistributionFeeOnBuy <= _MAX_FEE_LIMIT, "EtrondistributionFeeOnBuy can't exceed 25%!");
    liquidityFeeOnBuy = _liquidityFeeOnBuy;
    marketingFeeOnBuy = _marketingFeeOnBuy;
    EtrondistributionFeeOnBuy = _EtrondistributionFeeOnBuy;
    emit ChangeFeesForNormalBuy(_liquidityFeeOnBuy, _marketingFeeOnBuy, _EtrondistributionFeeOnBuy);
}

function changeFeesForWhiteListedBuy(uint8 _liquidityFeeOnBuy, uint8 _marketingFeeOnBuy, uint8 _EtrondistributionFeeOnBuy) external onlyOwner {
    require(_liquidityFeeOnBuy <= _MAX_FEE_LIMIT, "LiquidityFeeOnBuy can't exceed 25%!");
    require(_marketingFeeOnBuy <= _MAX_FEE_LIMIT, "MarketingFeeOnBuy can't exceed 25%!");
    require(_EtrondistributionFeeOnBuy <= _MAX_FEE_LIMIT, "EtrondistributionFeeOnBuy can't exceed 25%!");
    liquidityFeeOnWhiteListedBuy = _liquidityFeeOnBuy;
    marketingFeeOnBuyWhiteListed = _marketingFeeOnBuy;
    EtrondistributionFeeOnBuyWhiteListed = _EtrondistributionFeeOnBuy;
    emit ChangeFeesForWhiteListedBuy(_liquidityFeeOnBuy, _marketingFeeOnBuy, _EtrondistributionFeeOnBuy);
}

function changeFeesForNormalSell(uint8 _liquidityFeeOnSell, uint8 _marketingFeeOnSell, uint8 _EtrondistributionFeeOnSell) external onlyOwner {
    require(_liquidityFeeOnSell <= _MAX_FEE_LIMIT, "LiquidityFeeOnSell can't exceed 25%!");
    require(_marketingFeeOnSell <= _MAX_FEE_LIMIT, "MarketingFeeOnSell can't exceed 25%!");
    require(_EtrondistributionFeeOnSell <= _MAX_FEE_LIMIT, "EtrondistributionFeeOnSell can't exceed 25%!");
    liquidityFeeOnSell = _liquidityFeeOnSell;
    marketingFeeOnSell = _marketingFeeOnSell;
    EtrondistributionFeeOnSell = _EtrondistributionFeeOnSell;
    emit ChangeFeesForNormalSell(_liquidityFeeOnSell, _marketingFeeOnSell, _EtrondistributionFeeOnSell);
}

function changeFeesForWhitelistedSell(uint8 _liquidityFeeOnSell, uint8 _marketingFeeOnSell, uint8 _EtrondistributionFeeOnSell) external onlyOwner {
    require(_liquidityFeeOnSell <= _MAX_FEE_LIMIT, "LiquidityFeeOnSell can't exceed 25%!");
    require(_marketingFeeOnSell <= _MAX_FEE_LIMIT, "MarketingFeeOnSell can't exceed 25%!");
    require(_EtrondistributionFeeOnSell <= _MAX_FEE_LIMIT, "EtrondistributionFeeOnSell can't exceed 25%!");
    liquidityFeeOnWhiteListedSell = _liquidityFeeOnSell;
    marketingFeeOnWhiteListedSell = _marketingFeeOnSell;
    EtrondistributionFeeOnWhiteListedSell = _EtrondistributionFeeOnSell;
    emit ChangeFeesForWhitelistedSell(_liquidityFeeOnSell, _marketingFeeOnSell, _EtrondistributionFeeOnSell);
}
```

- Recommendation:
  - Considered as good tax deduction practice is buy and sell fees **combined** not to exceed 25%.

FOUND THREATS

⚠ High Risk

Owner can change amountOfTokensToAddToLiquidityThreshold value. If amountOfTokensToAddToLiquidityThreshold value is higher than the accumulated fees in the contract or 0, selling fails.

```
uint256 public amountOfTokensToAddToLiquidityThreshold = 10*10*18;
function updateAmountOfTokensToAddToLiquidityThreshold(uint256 _amountOfTokensToAddToLiquidityThreshold) external onlyOwner {
    amountOfTokensToAddToLiquidityThreshold = _amountOfTokensToAddToLiquidityThreshold * (10 ** _DECIMALS);
    emit UpdateAmountOfTokensToAddToLiquidityThreshold(_amountOfTokensToAddToLiquidityThreshold);
}

function _swapAndAddToLiquidity() private swapping {

    uint256 tokenAmountForLiquidity = amountOfTokensToAddToLiquidityThreshold;
    uint256 amountToSwap = tokenAmountForLiquidity.div(2);
    uint256 amountAnotherHalf = tokenAmountForLiquidity.sub(amountToSwap);
.............
}
```

Values check proposal:

```
function _swapAndAddToLiquidity() private swapping {

uint256 contractCurrentBalance = IBEP20(address(this)).balanceOf(address(this));
uint256 previousThreshold = amountOfTokensToAddToLiquidityThreshold;

if(amountOfTokensToAddToLiquidityThreshold >= contractCurrentBalance || amountOfTokensToAddToLiquidityThreshold <= 1*(10 ** _DECIMALS)) {
    amountOfTokensToAddToLiquidityThreshold = contractCurrentBalance;
}

uint256 tokenAmountForLiquidity = amountOfTokensToAddToLiquidityThreshold;
uint256 amountToSwap = tokenAmountForLiquidity.div(2);
uint256 amountAnotherHalf = tokenAmountForLiquidity.sub(amountToSwap);

.....................
    amountOfTokensToAddToLiquidityThreshold = previousThreshold;
}
```

**This is proposed code, not included in the contract.*

Informational

Owner can exclude address from fees.

```
function setIsExcludedFromFee(address account, bool flag) external onlyOwner {
    _setIsExcludedFromFee(account, flag);
    emit SetIsExcludedFromFee(account, flag);
}
```

Owner can withdraw any tokens from contract.

```
function withdrawEthInWei(address payable recipient, uint256 amount) external onlyOwner {
    require(recipient != address(0), 'Invalid Recipient!');
    require(amount > 0, 'Invalid Amount!');
    recipient.transfer(amount);
}

function withdrawTokens(address token, address recipient) external onlyOwner {
require(token != address(0), 'Invalid Token!');
require(recipient != address(0), 'Invalid Recipient!');

    uint256 balance = IBEP20(token).balanceOf(address(this));
    if (balance > 0) {
        IBEP20(token).transfer(recipient, balance);
    }
}
```

Owner can change reward distributor contract.

```
function updateRewardDistributor(address _rewardDistributor) external onlyOwner {
    require(address(rewardDistributor) != _rewardDistributor, 'Reward Distributor already exists!');
    rewardDistributor = IRewardDistributor(_rewardDistributor);
    _allowances[address(this)][address(rewardDistributor)] = _MAX;
    _allowances[address(rewardDistributor)][address(pcsV2Router)] = _MAX;
    _isExcludedFromFee[address(rewardDistributor)] = true;
    _excludeFromReflection(address(rewardDistributor));
    emit UpdateRewardDistributor(_rewardDistributor);
}
```

Owner can change max transaction limit, but can’t lower it than 1% of total supply.

```
function setMaxTxPercent(uint256 maxTxPercent) external onlyOwner {
    require(maxTxPercent != 0, "Can't set max tx percentage to zero!");
    maxTxAmount = _tTotal.mul(maxTxPercent).div(10**2);
    emit SetMaxTxPercent(maxTxPercent);
}
```

RECOMMENDATIONS FOR

GOOD PRACTICES

1. Consider fundamental tradeoffs
2. Be attentive to blockchain properties
3. Ensure careful rollouts
4. Keep contracts simple
5. Stay up to date and track development

JAILKWON

GOOD PRACTICES FOUND

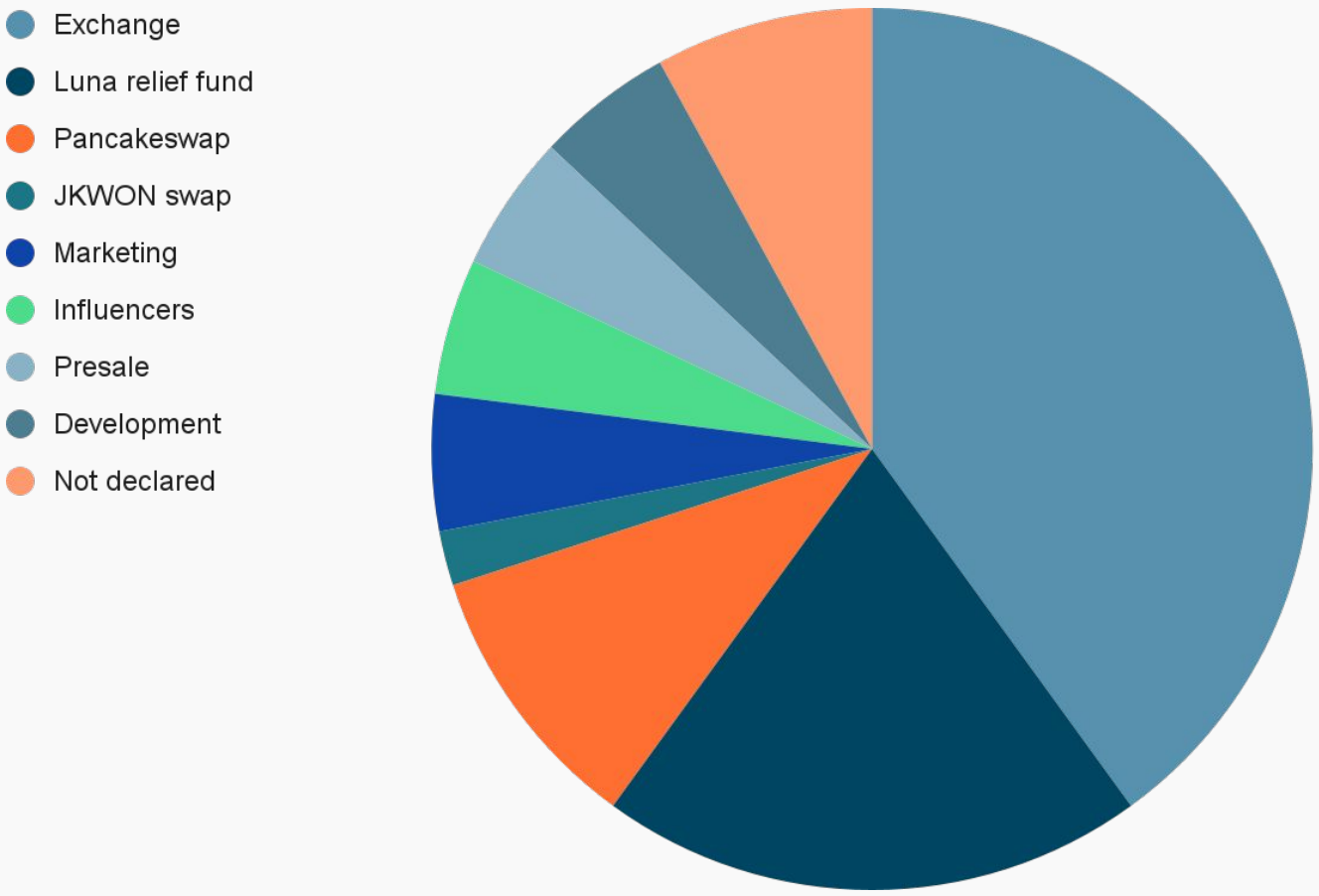
- ✔ The owner cannot mint new tokens after deployment
- ✔ The smart contract utilizes "SafeMath" to prevent overflows

TOKENOMICS

*The following tokenomics are based on the project's whitepaper and/or website:

- 5% - Presale
- 10% - Pancakeswap
- 2% - JKWON swap
- 5% - Marketing
- 40% - Exchange
- 20% - Luna relief fund
- 5% - Development
- 5% - Influencers

- **8% - Not declared**



THE TEAM

The team has privately doxxed to SPYWOLF by completing the following KYC requirements:

- ID Verification
- Video statement
- Video interview with devs
- Owner's wallet verification

KYC INFORMATION

Issuer

SPYWOLF

Members KYC'd

KYC Date

July 29, 2022

Format

Image

Certificate Link

https://github.com/SpyWolfNetwork/KYCs/blob/main/July/KYC_JAILKWON_0x24e537c31D18c926d96616898C62DF93a4a5c7E6.png

WEBSITE

Website URL
https://jailkwon.com/

Domain Registry
http://www.namecheap.com

Domain Expiration
Expires on 2023-05-22

Technical SEO Test
Passed

Security Test
Passed. SSL certificate present

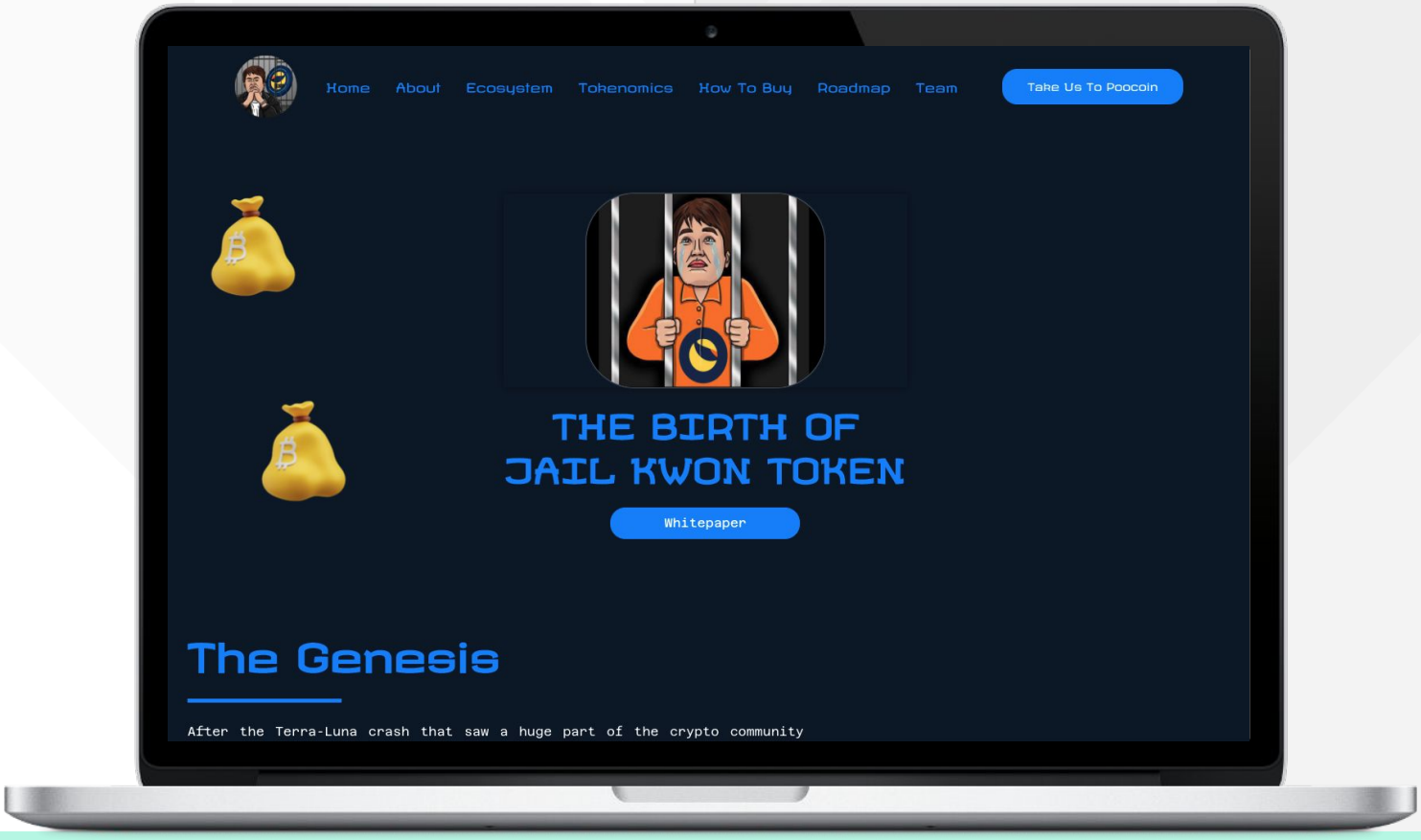
Design
Single page, appropriate color scheme.

Content
The information helps new investors understand what the product does right away. No grammar mistakes found.

Whitepaper
Yes. Explains the project's goals very well.

Roadmap
Yes, goals set without time frames.

Mobile-friendly?
Yes



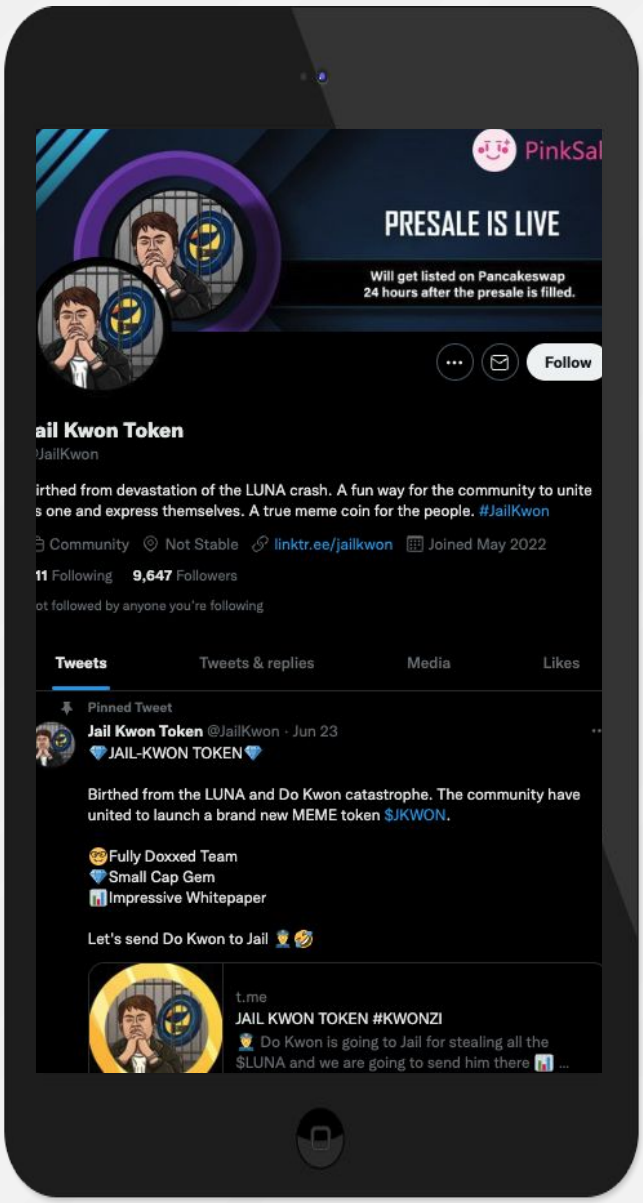
jailkwon.com

SOCIAL MEDIA

& ONLINE PRESENCE

ANALYSIS

Project's social media pages are active on a daily basis.



Twitter

@jailkwontoken

- 9 399 followers
- Active
- Daily posts

Discord

https://discord.gg/9ZzAeTdC

- 282 members
- ACtive

Telegram

@jailkwon

- 7 705 members
- Active members
- Active mods

Medium

- Not available

SPYWOLF

CRYPTO SECURITY

Audits | KYCs | dApps
Contract Development

ABOUT US

We are a growing crypto security agency offering audits, KYCs and consulting services for some of the top names in the crypto industry.

- **OVER 150 SUCCESSFUL CLIENTS**
- **MORE THAN 500 SCAMS EXPOSED**
- **MILLIONS SAVED IN POTENTIAL FRAUD**
- **PARTNERSHIPS WITH TOP LAUNCHPADS, INFLUENCERS AND CRYPTO PROJECTS**
- **CONSTANTLY BUILDING TOOLS TO HELP INVESTORS DO BETTER RESEARCH**

To hire us, reach out to
contact@spywolf.co or
t.me/joe_SpyWolf

FIND US ONLINE

SPYWOLF.CO

SPYWOLF.NETWORK

@SPYWOLFNETWORK

@SPYWOLFOFFICIAL

@SPYWOLFNETWORK

@SPYWOLFNETWORK

Disclaimer

This report shows findings based on our limited project analysis, following good industry practice from the date of this report, in relation to cybersecurity vulnerabilities and issues in the framework and algorithms based on smart contracts, overall social media and website presence and team transparency details of which are set out in this report. In order to get a full view of our analysis, it is crucial for you to read the full report.
While we have done our best in conducting our analysis and producing this report, it is important to note that you should not rely on this report and cannot claim against us on the basis of what it says or doesn't say, or how we produced it, and it is important for you to conduct your own independent investigations before making any decisions. We go into more detail on this in the disclaimer below – please make sure to read it in full.

DISCLAIMER:
By reading this report or any part of it, you agree to the terms of this disclaimer. If you do not agree to the terms, then please immediately cease reading this report, and delete and destroy any and all copies of this report downloaded and/or printed by you. This report is provided for information purposes only and on a non-reliance basis, and does not constitute investment advice.
No one shall have any right to rely on the report or its contents, and SpyWolf and its affiliates (including holding companies, shareholders, subsidiaries, employees, directors, officers and other representatives) (SpyWolf) owe no duty of care towards you or any other person, nor does SpyWolf make any warranty or representation to any person on the accuracy or completeness of the report.
The report is provided "as is", without any conditions, warranties or other terms of any kind except as set out in this disclaimer, and SpyWolf hereby excludes all representations, warranties, conditions and other terms (including, without limitation, the warranties implied by law of satisfactory quality, fitness for purpose and the use of reasonable care and skill) which, but for this clause, might have effect in relation to the report. Except and only to the extent that it is prohibited by law, SpyWolf hereby excludes all liability and responsibility, and neither you nor any other person shall have any claim against SpyWolf, for any amount or kind of loss or damage that may result to you or any other person (including without limitation, any direct, indirect, special, punitive, consequential or pure economic loss or damages, or any loss of income, profits, goodwill, data, contracts, use of money, or business interruption, and whether in delict, tort (including without limitation negligence), contract, breach of statutory duty, misrepresentation (whether innocent or negligent) or otherwise under any claim of any nature whatsoever in any jurisdiction) in any way arising from or connected with this report and the use, inability to use or the results of use of this report, and any reliance on this report. The analysis of the security is purely based on the smart contracts, website, social media and team.
No applications were reviewed for security. No product code has been reviewed.